



Plano de Projeto - F-ARM

1. Informações gerais do projeto

Informação	Detalhe
Nome do Projeto	F-ARM
Product Owner	Gabriela Barretto
Scrum Master	Gabriela Barretto
Time de Desenvolvimento	Gabriela Barretto
Data de Início	09/02/2025
Data de Término Prevista	19/12/2025

Data de Início: 09/02/2025

Data de Término Prevista: 19/12/2025

2. Seção de acordos entre aluno e parceiro

- A cada Sprint - Duas Semanas
- Apresentação do desenvolvimento realizado e feedback
- Remoto
- Acesso à fazenda e seus engenheiros agrônomos
- Nível de acesso a ferramentas - Moderado (Rede + Local para testes)

3. Objetivos do projeto

O **Agricultural Remote Monitoring (F-ARM)** visa o monitoramento remoto da qualidade ambiental em fazendas, com foco na análise do solo e do ar. A solução permite a coleta e transmissão de dados ambientais em tempo real para um aplicativo móvel intuitivo. Sensores RS485 e Analógicos serão conectados a módulos LoRa, que transmitirão os dados para um ESP32 coletor, responsável pelo envio via LoRaWAN para um gateway central. O gateway transmitirá os dados para a nuvem via **MQTT (AWS IoT Core)**, possibilitando que o aplicativo móvel exiba métricas em tempo real, fornecendo informações essenciais para otimizar a produtividade agrícola e gerar alertas preventivos sobre variações ambientais.

Além da coleta e exibição de métricas ambientais, o projeto incluirá uma etapa de **Data Science**, onde serão aplicadas técnicas de **regressão linear** para identificar relações entre o **Índice de Vegetação por Diferença Normalizada (NDVI)** – obtido através do portal [Embrapa SatVeg](#) – e os dados coletados pelos sensores. Essa análise permitirá correlacionar variáveis ambientais com a produtividade agrícola, fornecendo **insights inteligentes e intuitivos** que ajudarão na **tomada de decisão estratégica** e no **planejamento agrícola sustentável**.

4. Escopo do projeto - Backlog do produto:

- **Pesquisa inicial**
 - Tecnologias - Microcontroladores, protocolos de comunicação, sensores e ideias de arquiteturas para um projeto como este.
- **Arquitetura da solução**
 - Melhoria constante a cada sprint.
- **Análise financeira**
- **Canvas value proposition**
- **Matriz de risco**
- **Definição e Aquisição de Hardwares**
 - Definir como será a comunicação entre sensores e iot.
 - Definir quais sensores serão utilizados.
 - Definir o microcontrolador para nó coletor.
 - Definir Gateway LoRaWAN para infraestrutura.
 - Planejar alimentação (Solar/Bateria recarregável).
- **Configuração e Integração de sensores aos nós coletores**
 - Configurar e testar sensores (RS485 (Modbus RTU)).
 - Integrar os sensores ao transmissor (LoRa P2P (E32-433T30D)).
 - Criar firmware do sensor para envio periódico de dados.
 - Configurar ESP32 nó coletor para receber dados via receptor (LoRa P2P (E32-433T30D)).
 - Implementar filtro e validação dos dados recebidos no ESP32.
 - Testar as configurações do LoRa (Spreading Factor, potência, taxa de transmissão).
- **Configuração e Integração dos nós coletores com o Gateway via LoRaWAN**
 - Criar e testar gateway LoRaWAN.
 - Ajustar potência e alcance das antenas.
 - Implementar comunicação entre ESP32 e Gateway via LoRaWAN.
 - Criar firmware do ESP32 para transmitir os dados para o Gateway.
 - Testar e validar envio de dados contínuos.
 - Definir estratégia de backup.
- **Idealização do banco de dados**
 - Estruturar o banco de dados - Diagrama.
 - Diagramas de fluxo - Sensor - ESP32 - Gateway - AWS IoT - Lambda - RDS.
- **Configuração e Integração Gateway com Servidor em nuvem e BD**
 - Configurar AWS IoT Core para receber dados do Gateway via MQTT.
 - Configurar regras na AWS IoT Core para encaminhar dados para serviços na AWS (S3, Lambda).
 - Implementar API no AWS Lambda para processar os dados recebidos.
 - Criar validação e estruturação dos dados (integridade, redundância, e limpeza de dados).
 - Definir a estrutura do banco de dados na AWS (postgresql).
 - Criar conexão segura com PostgreSQL na AWS RDS.

- **Integração completa e validação da parte de Hardware**
 - Validar comunicação completa com envio de dados a um servidor.
 - Sensor - ESP32 (LoRa P2P) - Gateway - Servidor.
 - Validar autonomia energética dos componentes eletrônicos.
 - Executar testes o mais verossímil possível.
- **Análises Avançadas com Data Science (NDVI x Dados Ambientais)**
 - Coletar dados via API Embrapa (Penso por coordenada de cada sensor).
 - ETL dos dados coletados.
 - Pesquisar e escolher modelo de regressão linear.
 - Idealização de relatórios que agregam valor.
 - Geração de relatórios (Métricas de anos com maior produtividade).
 - Idealização da exposição dos relatórios.
- **Idealização do backend**
 - Pesquisa e definição de qual tecnologia utilizar (Node.js).
 - Documentação sobre as decisões e abordagens.
- **MVP do backed**
 - Implementar API REST para receber e processar os dados.
 - Criar endpoints (Buscar sensores, buscar últimas leituras).
 - Autenticação pelo menos JWT.
 - Criar WebSockets para transmissão de dados em tempo real.
 - Implementar para transmissão em tempo real.
 - Criar logs de monitoramento no CloudWatch
 - Testes e otimizações
- **Idealização do aplicativo mobile**
 - Pesquisa e definição de qual tecnologia utilizar (MAUI/Flutter).
 - Wireframe e documentação sobre as decisões e abordagens.
- **MVP Aplicativo Mobile**
 - Definir estrutura do app (Navegação).
 - Criar tela de login e autenticação.
 - Criar dashboard com gráficos e tabelas das leituras dos sensores.
 - Implementar mapa interativo mostrando sensores geolocalizados.
 - Conectar à API do backend via HTTP Requests.
 - Implementar WebSocket para atualizações em tempo real.
 - Testes e otimizações.
- **Integração completa e Testes**
 - Testar fluxo completo: Sensores - ESP32 - Gateway - API - Banco - App
 - Validar a comunicação entre os componentes. Backend e App.
 - Ajustar segurança e autenticação dos usuários.
 - Executar testes de carga e otimização do banco de dados.

5. Roadmap e Cronograma macro do projeto

Duração da Sprint: 2 semanas. 20 sprints no total.

- **Sprint 1:** Pesquisa inicial (Tecnologias, Microcontroladores, Protocolos de comunicação)
- **Sprint 2:** Arquitetura da solução, Value Proposition Canvas, Matriz de risco, Análise financeira
- **Sprint 3:** Definição e aquisição de hardwares
- **Sprint 4:** Configuração e integração de sensores aos nós coletores
- **Sprint 5:** Configuração e Integração dos nós coletores com o Gateway via LoRaWAN
- **Sprint 6:** Idealização do banco de dados
- **Sprint 7:** Integração do Gateway com o servidor em nuvem e banco de dados
- **Sprint 8:** Integração completa da parte hardware
- **Sprint (9-10):** Análises Avançadas com Data Science (NDVI x Dados Ambientais)
- **Sprint 11:** Idealização do backend
- **Sprint 12:** MVP do backend
- **Sprint 13:** Idealização do aplicativo mobile
- **Sprint 14:** MVP do aplicativo mobile
- **Sprint 15:** Integração completa e testes
- **Sprint 16-20:** Melhorias contínuas e otimizações para finalização do MVP

6. Premissas e restrições

Premissas:

- **Disponibilidade de Conectividade LoRaWAN**
 - Assume-se que a região onde o sistema será testado possui infraestrutura de LoRaWAN adequada ou que a instalação de um Gateway próprio será viável.
- **Operação em Ambientes Rurais**
 - O projeto será aplicado a áreas agrícolas, onde a conectividade tradicional (Wi-Fi ou 4G) pode ser limitada por isso o LoRaWAN.
- **Sensores e Componentes Eletrônicos Compatíveis**
 - Todos os sensores escolhidos utilizam comunicação **RS485 (Modbus RTU)** ou saída **analógica**, e são compatíveis com o ESP32 e módulos LoRa.
- **Consumo Energético Controlado**
 - Os sensores e dispositivos IoT funcionarão por longos períodos sem manutenção frequente, usando **bateria recarregável/solar**.
- **Gateway com Conectividade MQTT para Nuvem**
 - O Gateway LoRaWAN enviará os dados para a **AWS IoT Core** via **MQTT**, permitindo processamento e armazenamento.
- **Escalabilidade da Solução**
 - O projeto será modular, permitindo a expansão para novas métricas ambientais no futuro.
- **Aplicativo e Backend para Monitoramento**
 - O MVP incluirá um **aplicativo mobile** e um **backend em Node.js**, garantindo a visualização dos dados coletados.
- **Baixo Custo e Viabilidade para Pequenos e Médios Produtores**
 - O projeto será econômico e acessível para pequenos agricultores, sem necessidade de infraestrutura complexa.
- **Segurança e Integridade dos Dados**
 - O sistema garantirá a **segurança** dos dados coletados, transmitidos e armazenados.
- **Acesso aos Dados de NDVI (Data Science)**
 - Será possível obter dados de NDVI via **Embrapa SatVeg** ou outra fonte pública.
- **Capacidade de Processamento para Análises**
 - A AWS terá capacidade suficiente para executar os modelos de regressão linear.
- **Disponibilidade de Ferramentas de Análise**
 - Python e bibliotecas como Pandas, NumPy e Scikit-Learn serão utilizadas para as análises.

Restrições:

- **Custo Limitado**
 - O projeto deve ser **o mais econômico possível**, evitando sensores ou tecnologias excessivamente caras.
- **Infraestrutura Local Limitada**
 - Como é um projeto para áreas agrícolas, não pode depender de redes Wi-Fi estáveis ou energia elétrica contínua.
- **Baixo Consumo Energético**
 - Os nós coletores (ESP32 + LoRa) devem ter **autonomia energética**, funcionando com baterias ou painéis solares.
- **Foco na Conectividade LoRaWAN**
 - Outras formas de comunicação, como Wi-Fi e 4G, não serão priorizadas na PoC/MVP.
- **Limitações de Armazenamento e Processamento**
 - O **ESP32** e o **Gateway LoRaWAN** possuem processamento e memória limitados, exigindo otimização no firmware e na transmissão de dados.
- **Desenvolvimento dentro do Prazo do TCC**
 - O MVP deve ser entregue dentro do período do **TCC**, exigindo priorização de funcionalidades essenciais.
- **Dependência de Testes em Campo**
 - A validação depende de **testes reais em uma fazenda**, o que pode introduzir desafios logísticos e ambientais.
- **Segurança no Transporte de Dados**
 - A comunicação entre sensores, ESP32, Gateway e Nuvem deve ser segura, evitando perda ou manipulação dos dados.
- **Processamento e Armazenamento na Nuvem**
 - A AWS será utilizada para armazenamento e processamento, eliminando a necessidade de servidores locais.
- **Interface Simples e Intuitiva**
 - O aplicativo móvel deve ser simples e fácil de usar, permitindo que produtores rurais sem experiência técnica possam interpretar os dados.

7. Gestão de mudanças

A priorização do backlog no projeto **F-ARM** segue um modelo iterativo e incremental, permitindo ajustes conforme novas informações, necessidades dos stakeholders e desafios técnicos emergentes. A abordagem utilizada combina **Scrum** e **Kanban**, garantindo flexibilidade e eficiência.

Gestão de Mudanças no Projeto

Mudanças no **escopo, cronograma, custo ou outros aspectos** do projeto devem ser solicitadas formalmente e serão analisadas de acordo com os seguintes critérios:

- **Impacto no projeto:** Avaliação do impacto técnico, financeiro e operacional da mudança.
- **Valor agregado:** Benefícios diretos e indiretos da mudança para a solução.
- **Esforço necessário:** Complexidade da implementação e tempo necessário.
- **Viabilidade dentro do MVP:** Avaliação se a mudança pode ser incorporada sem comprometer os objetivos principais.

Processo de Priorização

Mudanças solicitadas seguem o fluxo abaixo:

- **Solicitação e análise:**
 - As mudanças podem ser sugeridas por qualquer membro da equipe ou stakeholder.
 - O impacto, esforço e valor agregado são avaliados pelo **Product Owner (PO)**.
- **Definição da Prioridade:**
 - **Urgente:** Correções críticas, falhas bloqueantes ou impactos significativos na entrega.
 - **Alta:** Funcionalidade essencial para o MVP e melhorias de usabilidade ou eficiência.
 - **Média:** Ajustes incrementais, refatorações e otimizações sem impacto imediato.
 - **Baixa:** Melhorias futuras, ajustes visuais ou funcionalidades adicionais para versões pós-MVP.
- **Atualização do Backlog:**
 - O **Product Owner** revisa e ajusta os PBI's, garantindo que as mudanças mais críticas sejam priorizadas.

- **Sprint Planning:**
 - Durante a **reunião de planejamento**, a equipe avalia o backlog priorizado/refatorado.
 - Alterações não planejadas são analisadas para determinar se devem entrar na sprint atual ou serem programadas para futuras iterações.
- **Desenvolvimento e Revisão:**
 - A equipe implementa as mudanças conforme o backlog priorizado/refatorado.
 - Em cada sprint, há um **momento de revisão** para apresentação das entregas e coleta de feedback dos stakeholders.
- **Ajustes Contínuos:**
 - O backlog pode ser atualizado continuamente com base no feedback das revisões e necessidades emergentes.
 - Mudanças que não afetam o MVP podem ser **realocadas para versões futuras**.

Feedback e Refinamento

- A cada **Sprint Review**, os stakeholders avaliam os avanços e fornecem feedback sobre as funcionalidades entregues.
- Caso novas necessidades surjam, elas passam pelo mesmo processo de análise e priorização.
- O backlog permanece **dinâmico**, garantindo que o projeto se adapte a mudanças sem comprometer prazos ou orçamento.

8. Gestão de riscos

Risco/Desafio	Impacto	Plano de Mitigação
Falha na comunicação LoRaWAN devido a interferências ou distância excessiva	Alto	Testar diferentes configurações de Spreading Factor , potência e tipo de antena. Utilizar repetidores se necessário.
Dificuldade de acesso aos dados	Alto	Implementar armazenamento local temporário no ESP32 ou Gateway para envio posterior quando houver conexão disponível.
Consumo energético alto nos nós coletores (ESP32 + LoRa)	Alto	Utilizar módulos de baixo consumo e implementar estratégias de deep sleep para otimizar a bateria. Testes com painéis solares e baterias de maior capacidade.
Falha em sensores (calibração imprecisa ou mau funcionamento)	Médio	Implementar validação de dados no firmware para detectar leituras inconsistentes. Ter sensores de backup para substituição rápida.
Problemas na transmissão de dados para a nuvem (MQTT para AWS IoT Core)	Alto	Implementar retry automático no ESP32 e Gateway. Criar logs para monitoramento e diagnóstico.
Custo elevado dos componentes e infraestrutura	Médio	Priorizar sensores e módulos custo-benefício sem comprometer a qualidade. Pesquisar fornecedores alternativos.
Dificuldade na integração entre hardware e software	Alto	Criar prototipagem incremental e testar os componentes separadamente antes da integração final.
Problemas na segurança dos dados transmitidos	Alto	Implementar criptografia na transmissão (AES para LoRa e TLS para MQTT) e autenticação JWT para o backend.
Falta de cobertura LoRaWAN na região de testes	Médio	Implantar um Gateway próprio ou testar soluções híbridas, como backup via Wi-Fi ou 4G.
Falta de usabilidade no aplicativo móvel	Médio	Fazer testes de usabilidade com feedback de produtores rurais para ajustes antes da entrega final.
Falta de testes em ambiente real	Alto	Criar um ambiente de testes controlado antes da implantação definitiva na fazenda.
Dificuldade na manutenção do hardware em campo	Médio	Projetar a instalação para permitir troca rápida de sensores e baterias sem necessidade de ferramentas complexas.
Escalabilidade limitada do sistema	Médio	Estruturar o backend e banco de dados de forma modular , permitindo expansão futura sem reestruturação completa.

Dificuldade em encontrar documentação e suporte para componentes usados	Médio	Escolher componentes com documentação completa e comunidade ativa para suporte técnico.
Dependência de serviços de terceiros (AWS, fornecedores de sensores, etc.)	Baixo	Avaliar alternativas open-source e garantir que o sistema tenha flexibilidade para troca de provedores.
Desempenho insatisfatório do modelo de regressão linear	Médio	Ajustar hiperparâmetros, testar outros modelos de regressão e usar técnicas como cross-validation para melhorar a performance.
Alto custo no processamento de análises na AWS (SageMaker, Lambda, RDS)	Médio	Otimizar scripts de análise para eficiência computacional e utilizar serviços gratuitos ou de baixo custo da AWS, como Lambda com camadas Python.
Dependência de scripts manuais para análises recorrentes	Médio	Automatizar o pipeline de ETL e regressões lineares com agendamento via AWS Lambda ou Step Functions.
Possível desalinhamento entre dados coletados e produtividade real	Médio	Validar o modelo com dados históricos da fazenda e ajustá-lo com base em novas coletas.

9. Gestão da comunicação

- Encontros semanais com o professor André Godoi.
- Livre contato via whatsapp e email com os colaboradores da empresa parceira.
- Comunicação direta com o professor André Godoi via Slack + Encontro semanal.

10. Melhoria Contínua

Revisão do documento a cada Sprint, com atualização mediante feedbacks e replanejamentos ao longo do desenvolvimento.